Materials Letters: X 7 (2020) 100046

Contents lists available at ScienceDirect

Materials Letters: X

journal homepage: www.elsevier.com/locate/mlblux

Tomographic image analysis of reinforcement distribution in composites using a flexible and material's specialist-friendly computational environment

Fernando Birra [a], Magda Encarnação [a,b], Adriano Lopes [c], Pedro Medeiros [a], Nuno Oliveira [d], Bruno Preto [a], Paulo Quaresma [a], Alexandre Velhinho [b,*]

[a] NOVA LINCS, Informatics Department, Faculty of Science and Technology, Universidade NOVA de Lisboa, 2829-516 Caparica, Portugal
[b] CENIMAT/i3N, Materials Science Department, Faculty of Science and Technology, Universidade NOVA de Lisboa, 2829-516 Caparica, Portugal
[c] ISTAR-IUL, Departamento de Ciências e Tecnologias da Informação (ISTA), ISCTE Instituto Universitário de Lisboa, 1649-026 Lisboa, Portugal
[d] Área Departamental de Engenharia Electrónica e Telecomunicações e de Computadores, ISEL - Instituto Superior de Engenharia de Lisboa, Rua Conselheiro Emídio Navarro, C, 1959-007 Lisboa, Portugal

ARTICLE INFO

Article history:
Received 31 July 2019
Received in revised form 4 February 2020
Accepted 13 April 2020
Available online 15 April 2020

Keywords:
Computed tomography
Metal-matrix composites
Problem Solving Environment

ABSTRACT

A computational environment - in the following called Problem Solving Environment (PSE) - dedicated to the analysis of tomographic images of composite materials is presented. The PSE current version is centered on reinforcement characterization and its main features are: (i) running on a desktop PC equipped with GPGPUs (General Purpose Graphical Processing Units); (ii) allowing a non-specialist in Computer Science to define visual programs that specify a sequence of processing steps; (iii) execution times compatible with an interactive use, due to the computational processing power of GPUs; (iv) the inclusion of visualization modules and the possibility of steering the computations through parameter changes.

1. Introduction

Researchers in the field of Materials Science are concerned with a wide range of characterization challenges. 3D geometrical characterization of the material's constituent phases, if correctly addressed, may reveal itself as a profitable tool to cope with such challenges [1], guiding the researcher towards the elucidation of specific issues concerning the amply recognized interdependency between the material's structure, processing effects, properties and applications [2]. 3D imaging, and particularly X-ray micro/-nanotomography (μ/nCT), constitutes a privileged way to gather information leading to a material's 3D geometrical characterization [3,4].

Geometrical information obtained from μ /nCT may relate to a specific constituent, concerning its morphology and other surface characteristics, spatial distribution (location-wise, as well as orientation-wise), volume fraction and assessment of interactions within elements of the reinforcement population. It can also be concerned with the interfaces between dissimilar constituents within the material: e.g. the volume and distribution, within an allegedly compact part, of pores relative to the solid [5–7]; or the topology of different constituent phases within the material [5,6,8,9].

Furthermore, *in situ* studies of solid/liquid interactions are of undoubted interest to the development and optimization of different processing techniques [7,10–18], while *in situ* characterization of internal damage holds a particular significance if knowledge is to be gained regarding material behaviour [19–30].

CT has been proven as a useful technique to study such questions in the particular cases of metal alloys, ceramics, metal- and polymer-foams, MMCs and fiber-reinforced ceramic-matrix composites (CMCs), biomaterials, etc. [4].

From a practical standpoint, however, one of the major problems concerning the practical use of μ/nCT concerns the large size of the datasets, and the corresponding massive computing power needed to cope with such large data. The tomographic reconstruction process, which transforms a set of raw projections into a stack of slices through the object providing a 3D volume for visualization and analysis, is very computer intensive. The tomographic image thus reconstructed corresponds to a 3D-matrix where each voxel is represented by an integer or a float; considering one byte per voxel and a sample of 600 × 1000 × 1000 voxels, we are faced with the need to store and manipulate around 600 Mbytes of information [31]. Thereafter, 3D analysis itself poses a lot of challenges.

* Corresponding author.
E-mail address: ajv@fct.unl.pt (A. Velhinho).

https://doi.org/10.1016/j.mlblux.2020.100046
2590-1508/

Some tools can be implemented as an extension of 2D counterparts, but others have to be approached differently. Adding an extra dimension also results in extra memory requirements and a clear need for high performance routines.

Therefore, the Materials Science specialist is very much in need of a dedicated software application which will enable him to:

• Visualize the data in 3D;

• Perform different image processing operations in order to remove from the image any artefacts present, or to exclude any objects not relevant to the ongoing analysis;

• Derive a tailored set of parameters capable of adequately describing the relevant features of any object studied in detail, or otherwise establishing a statistical description of the entire population of objects under consideration;

• Obtain an accurate, flexible and synthetic visualization of either the 3D dataset or a chosen graphical representation of its characteristic features.

Given the requirements above, it is clear that computational systems dedicated to the characterization of composite materials must include several modules:

• 3D image processing components: as described before, tomographic images need several steps of processing before one can extract the characteristics of the objects present in it. Processing is made by combining several operations in a pipeline, where one starts with the original image and each step produces a modified image that is the input of the next step. Some of the steps are computationally demanding, either from the CPU power needed either from the dimensions of main and secondary storage involved; another frequent aspect is the need for visual steering of the computations, where the adjustment of the some processing parameters is done through interactive visualization of the results produced.

• 2D and 3D visualizers: as implied by the nature of data and of the image processing steps described above, 2D and 3D visualization facilities are crucial in different parts of the processing. 3D large dataset visualization with good interactivity poses high demands to the graphics hardware and also requires CPU power to perform the needed calculations.

• Image archive: Each tomographic image can occupy several GigaBytes in main and secondary memory. There is also the necessity of storing and retrieving images corresponding to intermediate steps of processing. This need for disk space can be mitigated by the pipeline structure of the image processing programs

• Object characteristics extraction: The first step of the 3D image processing labels each object present with a unique identifier. In this step geometric characteristics of each object are extracted; this step has also high computational demands, but steering is not needed.

• Characteristics database: the characteristics extracted from a given sample are stored in a database; this information can be retrieved by information visualizers or exported in different formats for processing with several tools.

• Data visualization components: Data stored in the characteristics database can be presented in different ways to the Materials specialist.

Several commercial systems like Amira/Avizo (Group) have been mentioned for the processing of composite material tomographic image processing. Besides of cost factors, these systems include part of the functionalities described above, including the dataflow approach. Its main limitations come from the difficulty in integrating specially tailored processing modules in the system; this aspect implies long response times when applying some types of operation to data. As explained above, some of the processing steps are computationally demanding, and response times can only be reduced to values suitable for an interactive by use of parallel processing.

There are several references to use of parallel processing for the processing of tomographic data, starting with the build of the 3D image [32] to further processing steps [33].

In view of the limitations felt, the authors have undertaken the development of an alternative system, based on an open access software platform and specifically designed for materials characterization through tomography, hereon designated TomoGPU.

2. Experimental

According to the objectives stated above, two fundamental design choices were:

• to target hardware that, on one hand, would be powerful enough to handle the demanding computational tasks, and on the other hand would be accessible to teams with a limited budget. These constraints led to opt by running the application in a personal desktop computer equipped with a General Purpose Graphical Processing Unit (GPGPU) [34]. Running the application only on a local machine has the advantages of reducing the turnaround time, thus allowing an interactive use of the system.

• to base the development of the software in a toolkit for building problem solving environments (PSE). A PSE [35] is a software system where a set of processing algorithms dedicated to a particular area of science or engineering, and visualization and steering facilities are packaged in an application that can be easily used by a specialist in that particular area. This option allowed us to use an incremental approach to the development of the application, where already-existing components were combined with specially built modules in order to integrate new facilities in the system. Several toolkits for building these kind of systems exist; SCIRun [36] has been chosen due to it being a free and open software, with reasonable documentation, and being used in several areas of science; an example is described in [37].

The general organization of TomoGPU, is shown in Fig. 1. The user is first confronted with a SciRun window where, using modules fetched from a toolbox, he can build a processing sequence specifically adapted to the characteristics of the raw 3D image to be processed. In this task he is aided by a real-time visualization module, which shows the result effects of parameter changes in the processed image. Once the user is satisfied with a particular set of parameters, a binary 3D image is generated, which is then subject to a series of specialized modules.

One of these performs the labelling of the image, i.e., individually identifies all the relevant objects present in the image, thus providing data in the most appropriate form for subsequent operations. Such data may then be processed by a characterization module, which, based on the useŕs needs, extracts information concerning the objects' features, building a database, the contents of which may be visualised at will with the aid of a specific module.

Furthermore – even if beyond the scope of the present paper – the labelling module also generates a labelled 3D image, which can be viewed and manipulated by two other purpose-built modules.

A sequence of processing modules is built selecting each component from a menu on the left side. Some of the modules already existed in SCIRun (Johnson et al., 2013), but most of them are dedicated to the processing of tomographic images and are available under the menu item Tomo-GPU. In the following, we describe in detail each processing step and its implementation in our system.

The remainder of this work draws essentially on results obtained from tomographic experiments with an aluminium-matrix silicon carbide reinforced composite material [1,8]. Nevertheless, TomoGPU has also recently been tested with paleontological specimens, in the form of dinosaur egg shells [38].

In the following subsections, the operations described were performed in a desktop computer equipped with a dedicated

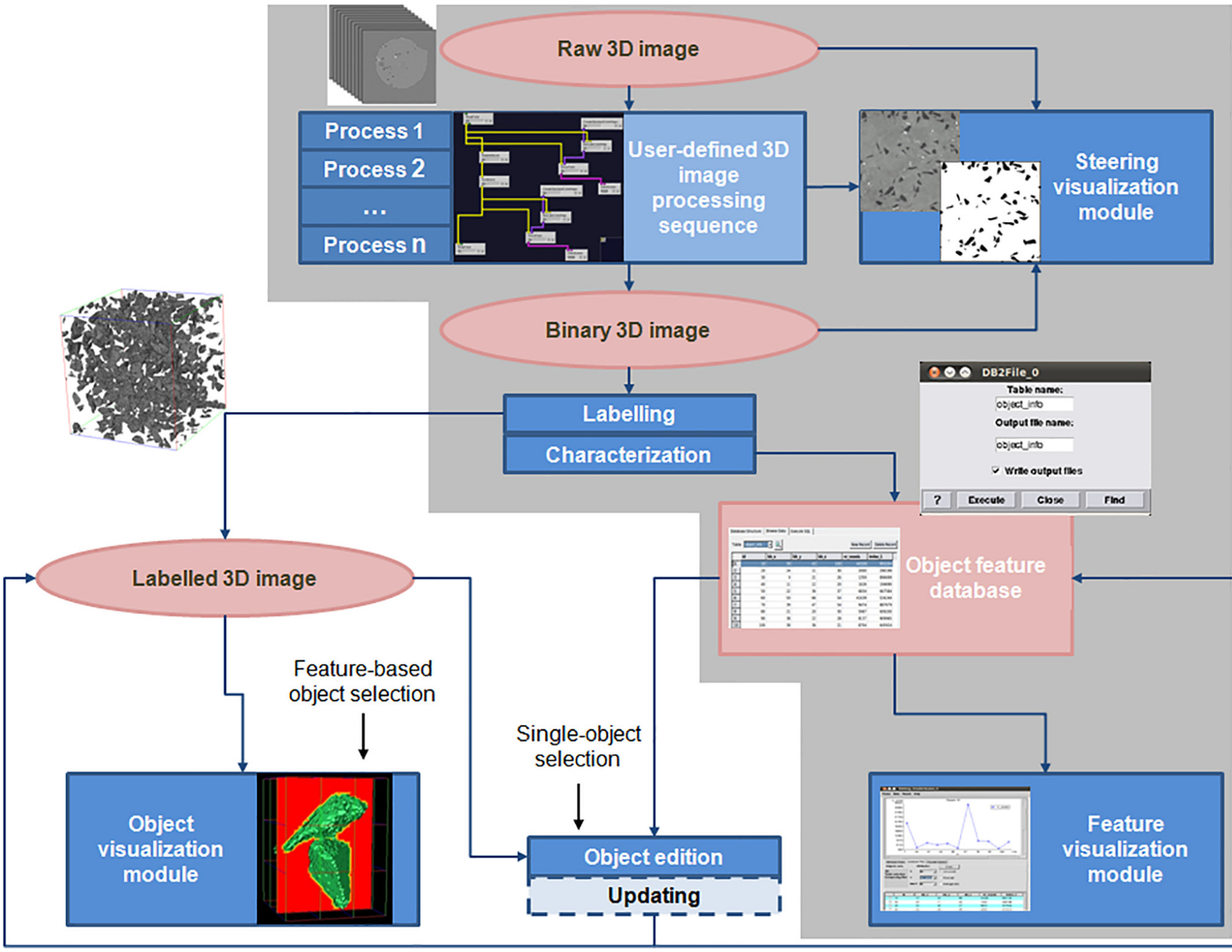

Fig. 1. General organization of the TomoGPU system, designed for tomographic imaging processing for the purposes of material's characterization.

accelerator card based on a nVidia GPU. The main hardware characteristics of the of the used system are Intel Xeon quad-core CPU E5506 at 2.13 GHz, 12 Gigabytes of main memory and 2 1-Terabytes disks; the GPU used is a nVidia Tesla C2050. Regarding software, Tomo-GPU was built over SCIRun version 4.5 running over Linux Ubuntu 12.04 and nVidia CUDA 6.5.

The relative computing power of the above described machine configuration notwithstanding, a materials science specialist could access a shared machine running TomoGPU from his/her laptop computer, through a graphical desktop sharing software, to process any tomographic images of interest.

The description made assumes that the matrix and the reinforcement objects have a significantly different density; this corresponds to a histogram presenting two peaks, where one corresponds to the matrix and other to the reinforcement objects. The system also includes facilities for the processing of samples the intervening phases are not so clearly separated in the histogram, due to having similar radiation absorption.

3. Results

The raw tomographic image is not readily adequate for object identification and measuring, so that it requires several steps of processing in order to eliminate noise and to accurately achieve proper object boundaries. Through segmentation, the greyscale image must then be transformed into its binary counterpart, where all voxels belonging to the objects present in the sample are turned into black, whereas the remaining voxels are coloured in white. Fig. 2 illustrates this stage of TomoGPU operation.

The image must first undergo a bi-segmentation operation. The input of this step is a 3D matrix input(i, j, k) where each voxel corresponds to a grey level between 0 (BLACK) and 255 (WHITE). The output is a three-value matrix output(i, j, k) defined according to

if(input(i, j, k) < L1) output(i, j, k) = BLACK

else if input(i, j, k) < L2) output(i, j, k) = GREY

else output(i, j, k) = WHITE

This fragment of code is included because it is representative of the data parallel nature of the algorithm. At the limit, a distinct thread will handle each voxel, allowing for an easy and efficient mapping to a GPU, where thousands of threads are supported with a very low overhead.

The bi-segmented image then is subject to the hysteresis module. The input of this step is a three-valued – BLACK, WHITE, GREY – 3D matrix and the output is again a 3D matrix with only BLACK and WHITE values. The conversion of grey voxels to black or white is made analysing the neighbours and deciding the voxeĺs colour according to a majority rule. This is an iterative process that stops when no grey voxels are found. This procedure also maps easily in the data-parallel characteristics of a GPU.

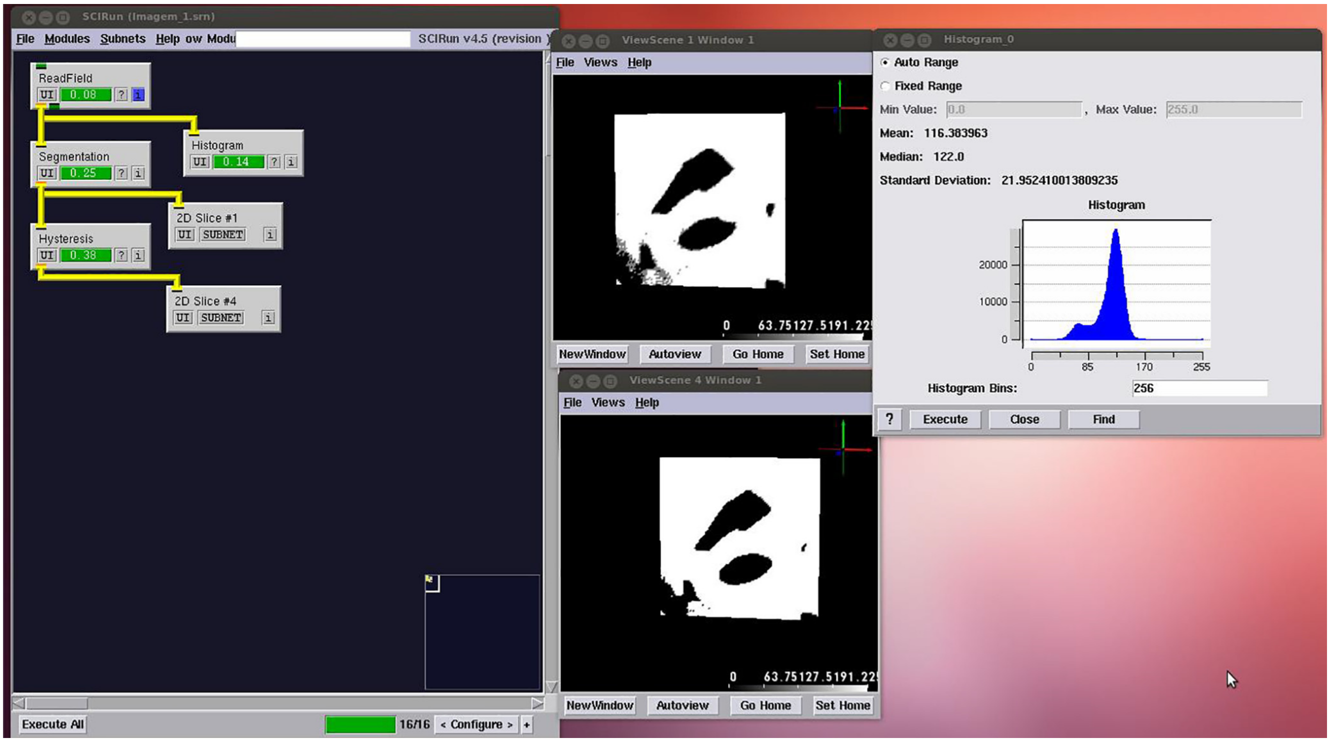

Fig. 2. Screenshot showing the processing sequence used to achieve adequate segmentation into a black and white image. The left window describes the visually built program sequence, including the bi-segmentation and hysteresis modules described below. Also shown are two windows used for visual comparison of the resulting image in different processing stages, as well as the histogram that guides the user's steering of the bi-segmentation process.

If one considers very large datasets (between 512 M and 1024 M voxels) the execution times of each of the two above operations are in the 1 s range, allowing an interactive use of the system.

The next step consists in labelling the image. The input of this processing step is a 3D matrix where each voxel is either 0 (BLACK) or 1 (WHITE); the outputs are two:

- A 3D matrix where each voxel has an integer label that identifies a distinct object in the sample; a distinct object comprises a set of contiguous black voxels and each distinct object has a unique label.
- A sequence of integers with the following format:
  - o Number of distinct objects
  - o For each object:
    - ■ Object's label
    - ■ Object's number of voxels
    - ■ A codification of each voxel ×, y, z coordinates

These two types of output are needed because they allow the optimization of subsequent processing operations. The 3D matrix format is more suitable for visualization of the entire sample; the sequence of objects output allows a more efficient process of individual object characterization.

Fig. 3 illustrates the appearance of the TomoGPU interface at this point.

The algorithm used for image labelling has been developed with the main objective of having small execution times; this is achieved by the use of GPUs and the design of the labelling algorithm has paid special attention to the match between each step and the hardware characteristics [39].

The execution times for a 1024x1024x1024 sample were around 6 s. This execution time is not good but tolerable in an interactive session; anyway, it is a huge improvement over previous implementations (more than 10x faster).

The different objects identified through the previous sequence are then characterized. Using the output of the labelling step, an analysis of each reinforcement object is performed. An object characterization from the geometric point of view is made, through the calculation of the volume, area, bounding box, inertial moment, etc. This analysis uses routines from the package CGAL [40]. A table in a database is created, where each reinforcement object corresponds to a line and each characteristic is a column. A Tcl/Tk-based [41] module allows a flexible visual presentation of the characteristics.

Fig. 4 shows the set of TomoGPU windows visible to the user during the labelling and statistical data visualization stage.

4. Discussion

The main objective of the TOMO-GPU system was to allow a non-specialist in Computer Science to define visual programs that specify a sequence of processing steps; when executing these programs the user should be able to steer the computations by interactively changing processing parameters and immediately observing the results of her/his changes through the inclusion of visualization modules.

The steps described in the previous section justify our claims about the functionality of the system. A Materials Science specialist could easily define a network of modules that process a 3D tomographic image of a composite material by choosing modules, either already existent either custom-developed, and interconnecting them. As shown in the example of Fig. 4, changing processing parameters is possible through a dedicated menu accessible through the button "UI" in each module. Each module also shows the time consumed in the process in the green indicator.

Regarding the claims of the interactivity of the system, they are justified by the fast execution times of the most demanding

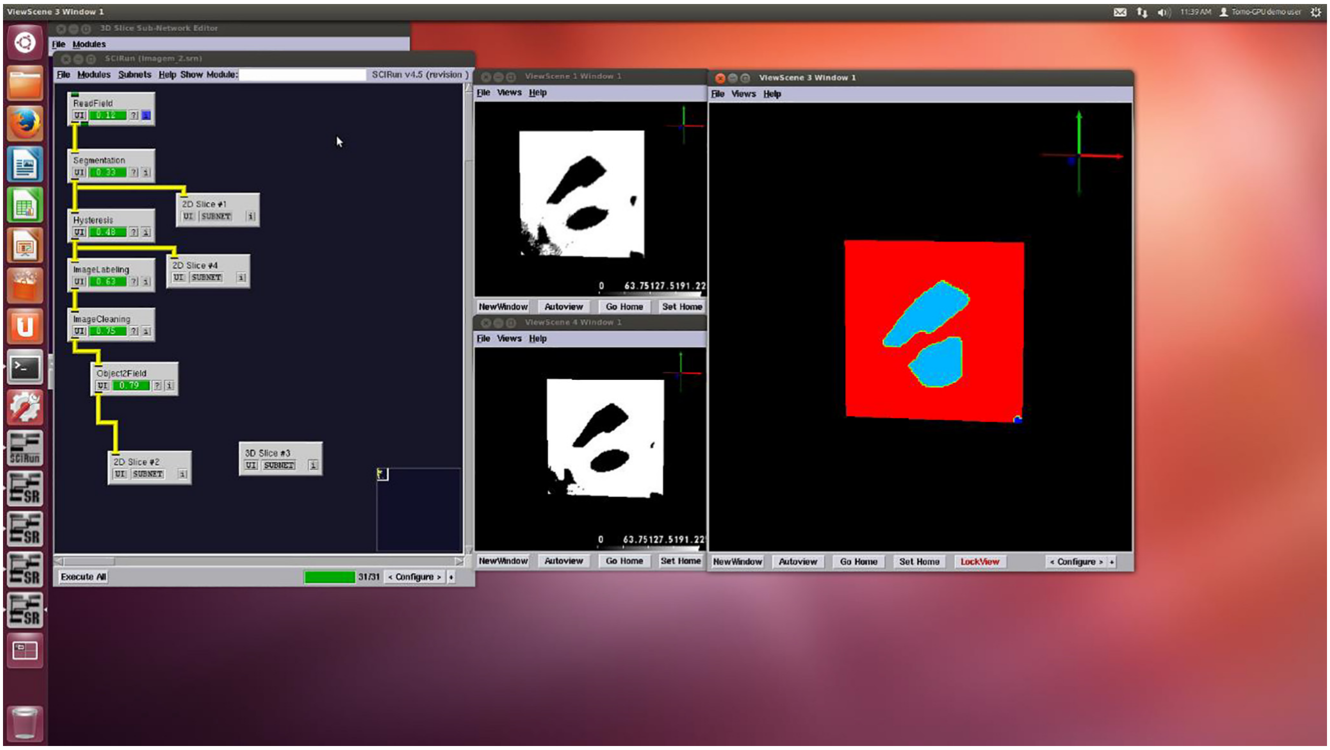

Fig. 3. Screenshot showing a more advanced stage of the processing sequence. The main addition concerns the image labelling module, which is tasked with the identification of the relevant objects present in the sample.

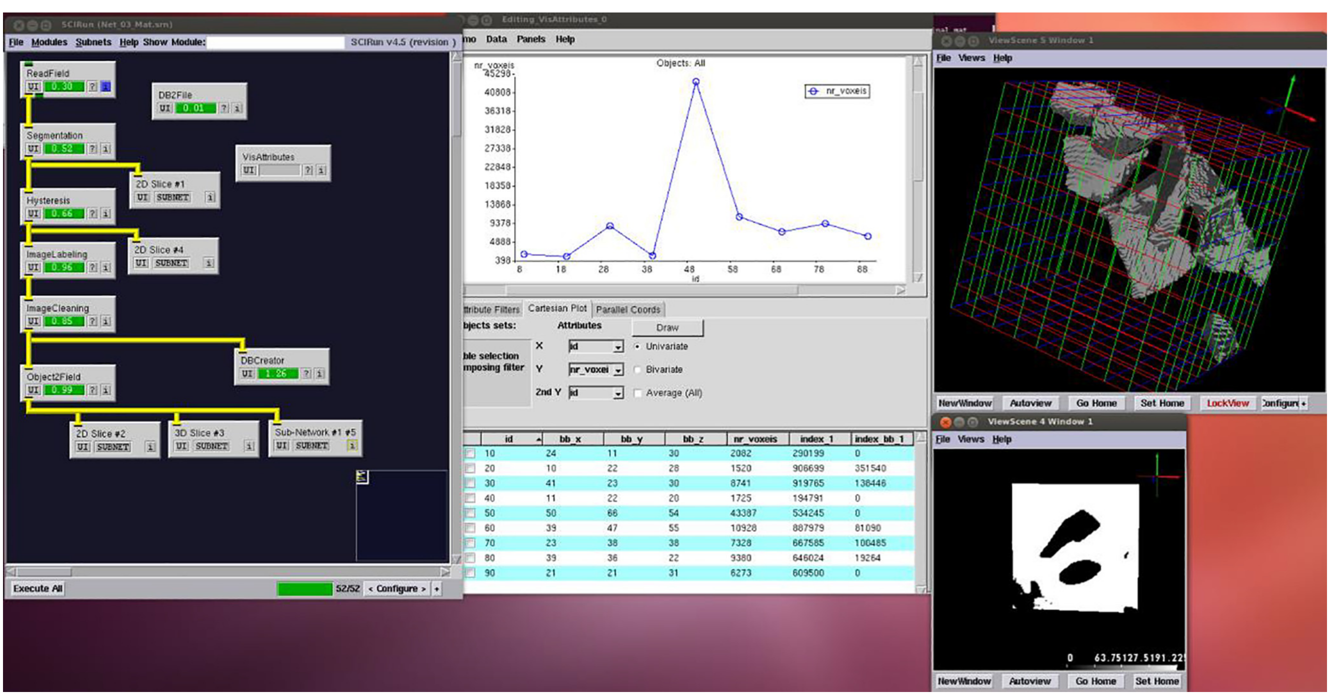

Fig. 4. The final stages comprise the DBCreator module, which generates a database containing each object's characteristics considered relevant by the user, as well as the VisAttributes module, which is tasked with producing the right-hand panel where selected statistics are displayed.

processing modules. Our experience shows that the most time-consuming processing step is "Image Labelling"; in [39] the algorithm used in this step is described and its performance extensively studied. Using a fine-tuned GPU implementation, execution times of 6 s are achieved for 1024x1024x1024 samples.

5. Conclusions

The TOMO-GPU system is built in a manner that allows a significant increase in the productivity of a researcher in the field of composite materials through expedite processing of tomographic datasets on an accessible hardware platform.

Using some built-in capabilities of SCIRun, along with specifically designed modules based on some enhanced algorithms, Tomo-GPU supports 3D image processing algorithms that have been tailored for efficient execution in GPGPUs, used to speedup lengthy processing steps.

Also database storage of object characteristics and interactive graphical object selection have been added to allow material analysis in detail.

Extensive imaging and information visualization modules are provided, both for image processing steering and final analysis of the results.

The system is currently reaching its final development stages, while tests have been performed through the treatment of particle-reinforced composites and paleontological specimens.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors wish to acknowledge FCT/MCTES for the financial support granted under contract PTDC/EIA-EIA/102579/2008 (Problem Solving Environment for Materials Structural Characterization via Tomography). Furthermore, A. Velhinho and M. Encarnação acknowledge funding support provided by FEDER funds through the COMPETE 2020 Program and National Funds through FCT - Portuguese Foundation for Science and Technology under the project number POCI-01-0145-FEDER-007688 Reference UID/CTM/50025, while F. Birra, A. Lopes, P. Medeiros, B. Preto and P. Quaresma acknowledge support provided under program PEst-OE/EEI/UI0527/2011. The authors are also grateful to Rui Martins for having made available tomographic images of paleontological specimens.

References

[1] A. Velhinho, F.M.B. Fernandes, S.C. Ferreira, L.A. Rocha, G. Vignoles, P. Cloetens, Adv. Sci. Technol. 45 (2006) 1109–1116.
[2] M.C. Flemings, R.W. Cahn, Acta Mater. 48 (2000) 371–383.
[3] E. Maire, J.-Y. Buffière, L. Salvo, J. Blandin, W. Ludwig, J. Létang, Adv. Eng. Mater. 3 (2001) 539–546.
[4] S.C. Mayo, A.W. Stevenson, S.W. Wilkins, Materials 5 (2012) 937–965.
[5] A. Velhinho, P.D. Sequeira, R. Martins, G. Vignoles, F.B. Fernandes, J.D. Botas, L. A. Rocha, Mater. Sci. Forum 423–425 (2003) 263–268.
[6] A. Velhinho, P.D. Sequeira, R. Martins, G. Vignoles, F.M.B. Fernandes, J.D. Botas, L.A. Rocha, Nucl. Inst. Methods Phys. Res. B 200 (2003) 295–302.
[7] Z. Yan, O. Guillon, S. Wang, C.L. Martin, C.-S. Lee, D. Bouvard, Appl. Phys. Lett. 100 (2012) 263107.
[8] A. Velhinho, F.B. Fernandes, J.D. Botas, G. Vignoles, X. Thibault, P. Cloetens, E. Boller, L.A. Rocha, Mater. Sci. Forum 492–493 (2005) 621–626.
[9] H.C. Watson, J.J. Roberts, Phys. Earth Planet. Inter. 186 (2011) 172–182.
[10] H. Toda, T. Yamaguchi, M. Nakazawa, Y. Aoki, K. Uesugi, Y. Suzuki, M. Kobayashi, Mater. Trans. 51 (2010) 1288–1295.
[11] H. Toda, K. Shimizu, K. Uesugi, Y. Suzuki, M. Kobayashi, Mater. Trans. 51 (2010) 2045–2048.
[12] Y. Hangai, Y. Ozeki, S. Kawano, T. Utsunomiya, O. Kuwazuru, M. Hasegawa, S. Koyama, N. Yoshikawa, Mater. Trans. 51 (2010) 548–552.
[13] G.L. Vignoles, C. Mulat, C. Germain, O. Coindreau, S. Delletrez, G. Chollon, in: Carbon 2009, Biarritz, France, 2009, pp. 10.
[14] N. Omura, Y. Murakami, M. Li, T. Tamura, K. Miwa, H. Furukawa, M. Harada, M. Yokoi, Mater. Trans. 50 (2009) 2578–2583.
[15] J. Adrien, E. Maire, N. Gimenez, V. Sauvantmoynot, Acta Mater. 55 (2007) 1667–1679.
[16] T. Ohgaki, H. Toda, I. Sinclair, J. Buffiere, W. Ludwig, T. Kobayashi, M. Niinomi, T. Akahori, Mater. Sci. Eng., A 406 (2005) 261–267.
[17] T. Ohgaki, H. Toda, I. Sinclair, J.Y. Buffière, W. Ludwig, T. Kobayashi, M. Niinomi, T. Akahori, Mater. Sci. Eng., A 427 (2006) 1–6.
[18] R.H. Mathiesen, L. Arnberg, P. Bleuet, A. Somogyi, Metal. Mater. Trans. A 37A (2006) 2515–2524.
[19] W. Hufenbach, R. Böhm, M. Gude, M. Berthel, A. Hornig, S. Ručevskis, M. Andrich, Compos. Sci. Technol. 72 (2012) 1361–1367.
[20] H. Toda, E. Maire, S. Yamauchi, H. Tsuruta, T. Hiramatsu, M. Kobayashi, Acta Mater. 59 (2011) 1995–2008.
[21] A.E. Scott, M. Mavrogordato, P. Wright, I. Sinclair, S.M. Spearing, Compos. Sci. Technol. 71 (2011) 1471–1477.
[22] C. Chateau, L. Gélébart, M. Bornert, J. Crépin, E. Boller, C. Sauder, W. Ludwig, Compos. Sci. Technol. 71 (2011) 916–924.
[23] A. Borbély, K. Dzieciol, F. Sket, A. Isaac, M.d. Michiel, T. Buslaps, A.R. Kaysser-Pyzalla, Journal of Materials, 63 (2011) 78-84.
[24] H. Toda, H. Oogo, K. Uesugi, M. Kobayashi, Mater. Trans. 50 (2009) 2285–2290.
[25] A. Steuwer, L. Edwards, S. Pratihar, S. Ganguly, M. Peel, M.E. Fitzpatrick, T.J. Marrow, P.J. Withers, I. Sinclair, K.D. Singh, N. Gao, T. Buslaps, J.Y. Buffière, Nucl. Instrum. Methods Phys. Res., Sect. B 246 (2006) 217–225.
[26] K.H. Khor, J.Y. Buffiére, W. Ludwig, I. Sinclair, Scr. Mater. 55 (2006) 47–50.
[27] E. Ferrié, J.-Y. Buffière, W. Ludwig, A. Gravouil, L. Edwards, Acta Mater. 54 (2006) 1111–1122.
[28] L. Daelemans, S. van der Heijden, I. De Baere, H. Rahier, W. Van Paepegem, K. De Clerck, Compos. Sci. Technol. 124 (2016) 17–26.
[29] H. Rolland, N. Saintier, G. Robert, Compos. B Eng. 90 (2016) 365–377.
[30] S.M. Sisodia, S.C. Garcea, A.R. George, D.T. Fullwood, S.M. Spearing, E.K. Gamstedt, Compos. Sci. Technol. 131 (2016) 12–21.
[31] T. Cadavez, S.C. Ferreira, P. Medeiros, P.J. Quaresma, L.A. Rocha, A. Velhinho, G. Vignoles, Int. J. Tomo. Stat. 14 (2010) 3–15.
[32] N. Neophytou, F. Xu, K. Mueller, in: SPIE Conference on Medical Imaging 2007, San Diego, USA, 2007.
[33] J.A.I. Gregor Von Laszewski , Ian Foster , John Bresnahan , Carl Kesselman , Mei Su , Marcus Thiebaux , Mark L. Rivers , Steve Wang , Brian Tieman , Ian Mcnulty, in: Ninth SIAM Conference on Parallel Processing for Scientific Computing, 1999.
[34] J. Nikols, W. Daly, in: IEEE Micro, 2010.
[35] . Gallopoulos, N.H. E, J.R. Rice, IEEE Computational Science & Engineering, 1 (1994) 1123.
[36] S.G. Parker, C.R. Johnson, in: IEEE/ACM Supercomputing Conference 1995, San Diego, USA, 1995.
[37] R.S. McLeod, in: IEEE International Symposium on Biomedical Imaging: Nano to Macro, 2004.
[38] R.M.S. Martins, R.A. Castanhinha, O. Mateus, R. Araújo, F. Beckmann, N. Schell, Ciência & Tecnologia dos Materiais 24 (2012) 36–39.
[39] B. Preto, F. Birra, A. Lopes, P. Medeiros, Int. J. Creat.Interf. Comput. Graphics 4 (2013) 40–56.
[40] CGAL 4.8.1 Manual: http://doc.cgal.org/latest/Manual/packages.html; accessed 25th July 2016.
[41] B. Welch, K. Jones, J. Hobbs, Practical Programming in Tcl and Tk, 4th ed., Prentice-Hall, 2003.